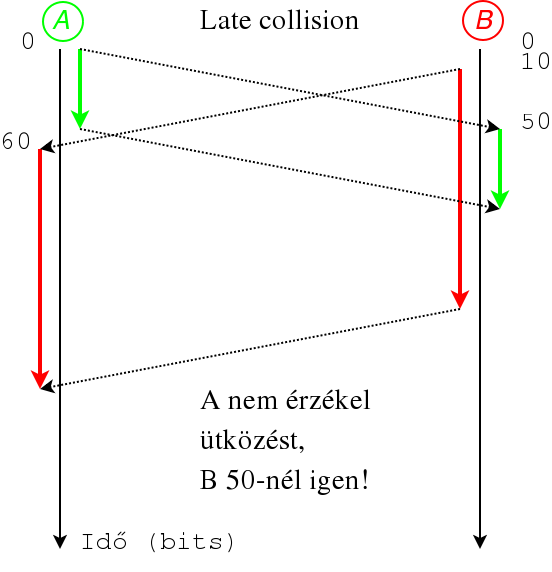
Villámkérdések

* Ephemeral port
  + A szerver vagy kliens által véletlenszerűan választott port. Általában 1024<, de különböző tartományok léteznek.
* CIDR
  + Classless Interface Domain
  + megszünteti az IP cím osztályokat, ezzel vezetve a lassabb IP cím elavuláshoz
* RFC
  + Request For Comment
  + Internetes szabványok gyűjteménye
* IANA
  + Internet Assigned Numbers Authory
  + különböző internetes paraméterek (pl IP) kiosztásáért felelős
* IAB
  + Internet Architecture Board
  + Internet fejlődését felügyeli szervezet (szabványok) - RFC
* RIPE NCC
  + Francia neve van lol
  + Az európai IP címek osztásáért felelős
* Unicast/multicast/broadcast/anycast
  + egycímzett/több címzett/mindenki címzett/bárki lehet címzett (mentők)
* Simplex/duplex/half duplex
  + egyirányú/kétirányú/felváltott kétirányú
* CSMA/CD
  + CS = Carrier Sense
    - figyelem hogy ne beszéljek ha más ad
  + MA = Multiple Acces
    - mindenkit figyelek, de csak a saját csomagjaim érdekelnek
  + CD = Collision Detection
    - ütközésnél 32 zajbit, backoff delay, exponential backoff
* Late collision
  + nem lehet észrevenni az ütközést az egyik oldalon
  + korlátozni kell a kábelhosszt, meg a repeaterek számát
  + 
* Exponential backoff
* Little endian/Big endian
  + legkisebb helyiértékű az első nagy az utolsó
  + legnagyobb az első legkisebb az utolsó
* Spanning tree
  + Kör van a topológiában: csomagok körbe-körbe járnak egyéb gondok
  + Spanning Tree - feszítő fát keresek, aszerint járok el, többet is használhatok
* VLAN
  + Virtual LAN
  + egy ethernetet logikailag több részre osztunk
  + előnye: biztonság
* Loopback interfész
  + ha egy eszközön van a kliens és a szerver ~ localhost
  + 127.0.0.1
* MTU
  + Maximum Transition Unit
  + max amit áttudunk küldeni két hálózat között
  + általában 1500 byte
* TOS
  + Type Of Service
  + Milyen szolgáltatásokat tud a csomagunk
  + a hálózaton belüli továbbítást befolyásoló mező
* TTL – IP csomagnál, DNS rekordnál
  + Time To Live
  + IP-nél minden csomag csökkenti eggyel, ha 0 eldobjuk
  + nem kering végtelen ideig
* Netmask
  + Egy IP-hez képest egy tartomány.
  + Egy IP rajta van e a hálózaton?
* NAT
  + Network Adress Translation
  + Hálózaton belüli címet átalakítjuk, hogy kívül is alkalmazható legyen
  + Belső gépek közti kommunikáció miatt
* ARP cache poisoning
  + ARP cachébe hamis MAC cím kerül -> Man in The Middle
* Path MTU discovery
  + ICMP vezérlő üzenet
  + mi a legkisebb MTU a cél felé
* ICMP redirect
  + ha a router tudja, hogy van nála egy jobb szól
  + csak Gatewaytől!
* AS
  + Autonóm Rendszer
  + Routing szempontjából egy önálló entitás
  + azonosító: AS number, RIPE osztja
  + S fajták:
    - Stub: egy másik AS fele van kapcsolata
    - Multihomed: több AS fele van kapcsolata
    - Transit: nem csak a saját forgalmát bonyolítja
* UDP lite
  + Multimédiás kommunikáció esetén nem érdekel ha elvesztünk pár csomagot
  + a sérült is jobb mint a semmi
* SOA, NS, A, PTR, MX rekord
  + DNS Resource Record tagjai
  + Start of Authority
    - DNS zóna adatok
    - sorozatszám, időzítések
  + Nameserver
    - meghatározza, hogy melyik nameserverhez kell intézni adott lekéréseket
  + Adress
    - az állomás névhez tartozó IP
  + Pointer
    - HIÁNY
  + Mail Exchanger
    - levél célállomás
* FQDN
  + Fully Qualified Domain Name
  + klasszikus értelemben vett weblap cím, teljes elérési útvonal
* Primary (master), secondary (slave) DNS szerver
  + Egy zónát több szerver is kiszolgálhat
  + Van master és slave
  + primary adatbázisből vesz infőt
  + a slave-k a master alapján frissülnek
* Caching only DNS szerver, autoritatív név szerver
  + aut névszeró:egy zóna felelőse, egy ágé
  + ha nem autoritás, akkor caching only - ekkor nincs különösebb dolga, csak folyton lekéréseket hajt végre, amiből saját cachéjét firssíti
* DNS cache poisining
  + DNS cachét manipulálják
  + pornhub helyet ducktv oldala jön be és ellopják az adataid
* TCP syn flood támadás
  + egy gép folyton SYN flages csomagokat küld random portokról/IP-kről
  + a támadó gép küldené az ACKokat, de nem bír velük -> DOS
* TCP aktív/passzív open
  + TCPkapcsolat építés 3 lépése:
    - kezdeményező SYN-t küld
    - Jön egy ACK és egy SYN
    - Visszamegy az ACK
  + Kezdeményező: aktív open
  + elszenvedett: passzív open
* TCP aktív/passzív close
  + kezdeményező küld egy FIN-t B-nek
  + kap egy ACK-ot
  + ha már B sem akar küldeni visszaküld egy FIN-t
  + és a kezdeményező küld egy ACK-ot
  + ezután megszánik a kapcsolat.
  + Kezdeményező: aktív close
  + Elszenvedett: passzív close
* TIME\_WAIT (2MSL wait) állapot
  + Az aktív closet végrehajtó utolsó állapotánál
  + már megkapta a FIN-t, és elküldte az ACK-ot
  + de mi van ha az ACK elveszik, ezért vár 2MSL ideig utána zár
  + MSL: ennyi ideig lehet egy szegmens a hálózaon
* TCP reset támadás
  + A és B kapcsolatban van
  + E bejuttatt egy Resetet és így felbontja a kapcsolatot
  + IP-ket portokat, sequence numbert tudni kell hozzá
* TCP retransmission timer (RTO)
  + Ha RTO ideig nem jön ACK egy csomagra, újra kell küldeni, mert elveszett
  + egy időérték - Round Time Tripből (idő amíg odaér + ACK ideje) számoljuk
* Delayed ack
  + Nem azonnal küldünk ACK-ot
  + várunk még, hátha jön még adat
  + vagy hátha mi küldünk, amivel tudunk egyben nyugtázni is (piggy back)
* Slow start
  + nem csak a fogadó, hanem a küldő is flow controlt alkalmaz
  + kiegyensúlyozza a hálózat sebességét
  + lassan növeljük a csomagokat amiket küdlünk, a hálózat max kapacitásáig
  + congestion window - reciver window - egyensúly
* Persist timer
  + HIÁNY
* Passzív ftp
  + Az adatkapcsolatot nem a szerver nyitja hanem a kliens
  + PASV paranccsal
  + szerver: PORT NR
  + kliens ephemeral portja és a kapott port között nyílik kapcsoalt
* UA, MTA
  + User Agent: a felhasználó mail programja
  + Mail Transfer Agent: a leveleket továbbító mail program
* DHCP lease
  + DHCP dinaimikus IP osztása, időre adom
  + IP + kliens + intervallum = DHCP lease
* PXE (Preboot Execution Environment)
  + turbós DHCP Inteleken
* Proxy ARP?
  + ha a másik hálózatban van akit keresünk
  + broadcastre nincs válasz
  + amúgy sincs értelme ha másik hálózatban van
  + a router azt hazudja magáról hogy ő az, és rajta keresztül küldünk
  + Default Gateway: ha eleve tudom, hogy más hálózatban van, neki küldöm és rábízom magam.
* Hogyan működik az IP csomag fragmentálás?
  + ha az IP csomag nagyobb mint az MTU, akkor darabolni kell
  + csak akkor tudjuk ha a fragment bit áll
  + a darabokat a fragment offszet érték szerint sorba rendezhetjük
  + ID azonos
  + ha nem áll a fragment bit ICMP error
* Miért felejtenek és mit felejtenek switch-ek?
  + MAC címeket időnként elfelejtik
  + dinamikus a hálózat, lehet valaki nem tér vissza
  + pl Meki
* Mi a különbség hub és switch közt?
  + Hibás adatot a switch nem küld tovább
  + a HUB kb egy repater nem tanul címeket
  + a switch fejlettebb
* Miért felejtik el az eszközök az ARP bejegyzéseiket?
  + dinamikusság miatt, hogy észrevegyék a hálózaton a változásokat
* Hogyan működik a traceroute?
  + egyre növekvő TTL-el küld csomagot
  + majd vár választ
  + Time exceed, vagy megérkezést
  + rendszer diagnosztika, hálózat térképezés
* Hogyan működik a ping?
  + küldünk egy echo üzenetet, majd a céltól változás nélkül visszakapja
  + haszna, hogy le tudjuk ellenőrizni hogy valami elérhető e a hálózaton
  + útidőt számol és adatveszteséget
* Miért tartalmazzák az ICMP hibaüzenetek a hibát kiváltó csomag egy részét is?
  + Milyen IP címről jött a hiba, melyik porton -> magyarul hol a hiba?
* Mire vezetne, ha icmp hiba üzenetre icmp hiba üzenet lehetne a reakció?
  + agyon terhelné a hálózatot, végtelenségig küldözgethetnék egymásnak
* Mit értünk routing protokolloknál konvergencián?
  + routing táblák küldése, utak optimalizálása
  + egy időután beáll egy egyensúlyi állapot
* Mi a kockázata annak, ha bárhonnan elfogadunk RIP üzeneteket?
  + HIÁNY
* Miért fontos, hogy a root név szerverek mindig elérhetőek legyenek?
  + ha nem elérhető a cachéból kiesett címeket nem fogjuk tudni elérni
* Mire szolgál és hogyan működik az inetd daemon?
  + unixokon
  + egyszerű szolgáltatások indítása (pl. echo)
  + standard input és output a TCP lesz
* Hogyan zajlik egy TCP kapcsolat felépítés?
  + TCPkapcsolat építés 3 lépése:
    - kezdeményező SYN-t küld
    - Jön egy ACK és egy SYN
    - Visszamegy az ACK
* Hogyan zajlik egy TCP kapcsolat bontás?
  + kezdeményező küld egy FIN-t B-nek
  + kap egy ACK-ot
  + ha már B sem akar küldeni visszaküld egy FIN-t
  + és a kezdeményező küld egy ACK-ot
  + ezután megszánik a kapcsolat.
* Mitől függ a TCP retransmission timeout?
  + RTT-től (csomag elküldésétől az ACK megérkezéséig tartó idő)
  + bRTT = a \* bRTT + (1-a) \* mRTT
  + RTO = b \* bRTT
  + bRTT: becsült RTt
  + mRTT: mért RTT
* FTP-nél/SMTP-nél hogyan csoportosítjuk a szerver hibakódokat?
  + xyz valami szöveg
  + HIÁNY
* Miért okoz gondot a tűzfalakon az aktív ftp?
  + ne engedjünk be olyan dolgokat, amiket más kezdeményez / egyből megvágják
* Hogyan zajlik egy SMTP kapcsolatfelépítés?
  + HELO fasz.szerver.fing
  + MAIL FROM: <jopuki@maki.suni>
  + RCPT TO: <todor@kolonia.tacsi>
  + DATA
  + szoveg
  + .
  + QUIT
* Mi az a looking-glass?
  + Az interneten elszórva http felületen lekérdezhető routerek
  + Diagnosztikai eszköz
* Mire való a whois szolgáltatás?
  + megadja egy domainhez tartozó IP-t, egyéb hasznos infók amik egy embert érdekelhetnek
* Egy TCP kapcsolatnál a receiver window értéke 1000, a congestion window (cwnd) 1500. Mekkora csomagot küldhetünk?
  + 1000-es különben nem tudná fogadni gondolom én
* Milyen üzenetekben láthatunk HTTP cookie-kat? Mire szolgálnak?
  + pl sok banner tartalmazza
  + a cookiek segítségével a szerver bekukucskál a gépünkbe
  + információcsomag
  + legtöbbször azonosítás, bevásárló kosár megtartására, nyomonkövetésre használják
  + gyakran reklámokkal jön -> célzott reklámokat akarnak
  + ha rossz kezekbe kerül szopás : XSS
* Mire való az SSL/TLS?
  + Netscape találmány
  + Secure Socket LAyer
  + nyilvános kulcsú titkosítás
  + védelem a böngészés felett
  + lehallgatás megakadályozása